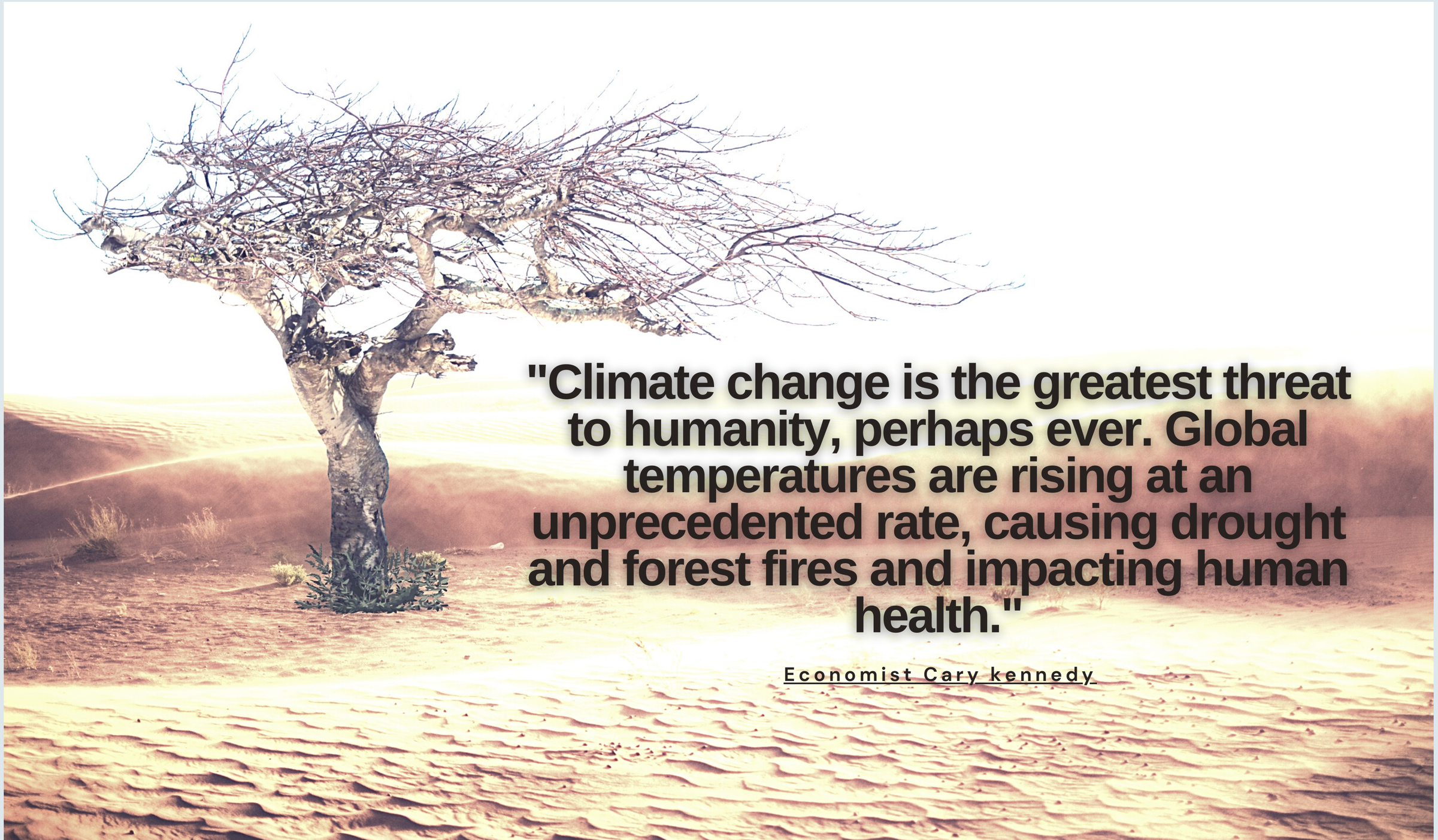
DROUGHT PREDICTION

Machine Learning Model

FINAL PROJECT

A presentation by Roxana Ugaz, Shreestina Tamrakar and Masi Mapiye

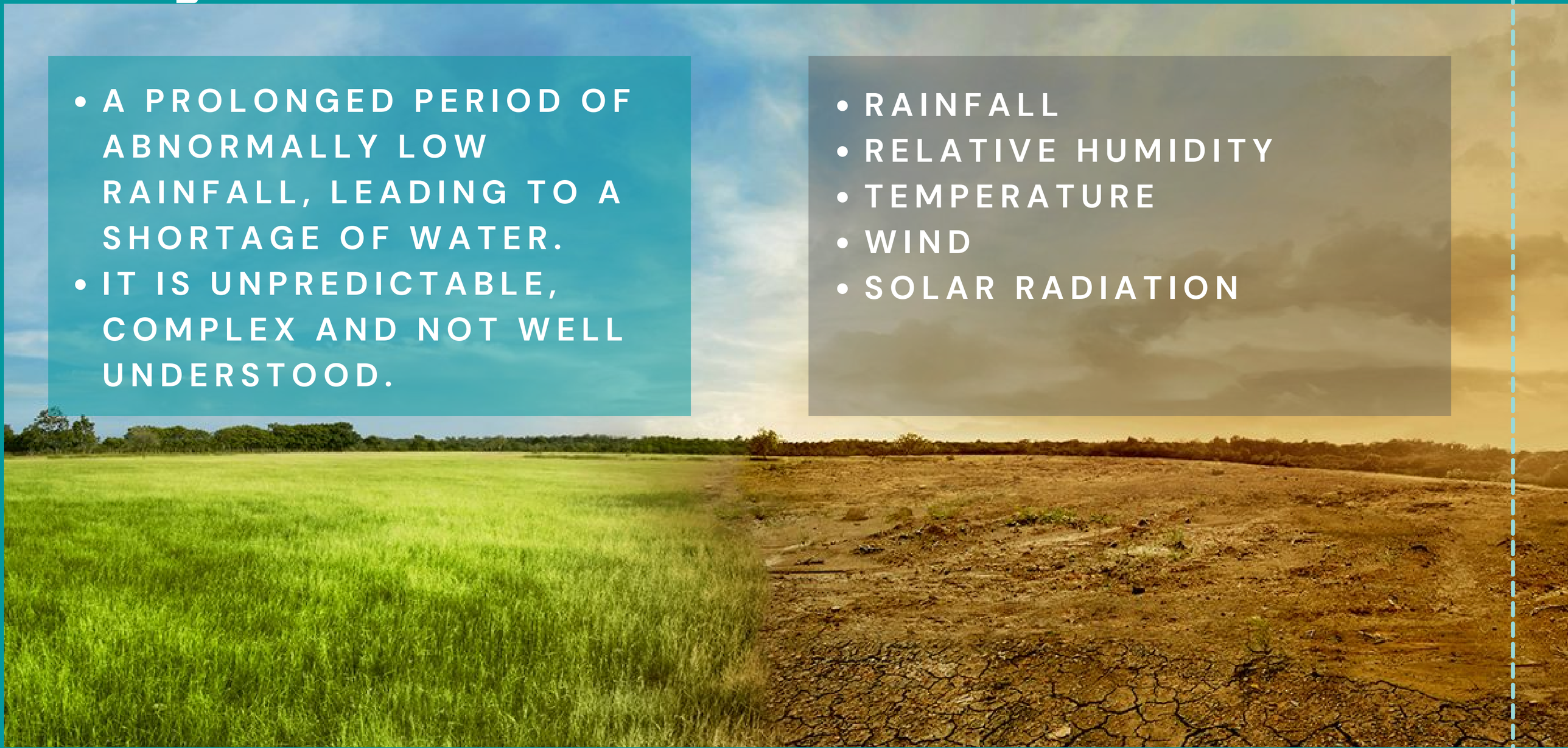

"Climate change is the greatest threat to humanity, perhaps ever. Global temperatures are rising at an unprecedented rate, causing drought and forest fires and impacting human health."
Economist Cary kennedy

What is drought?

- A PROLONGED PERIOD OF ABNORMALLY LOW RAINFALL, LEADING TO A SHORTAGE OF WATER.
- IT IS UNPREDICTABLE, COMPLEX AND NOT WELL UNDERSTOOD.

Factors affecting occurence and frequencies

- RAINFALL
- RELATIVE HUMIDITY
- TEMPERATURE
- WIND
- SOLAR RADIATION



Research Scope

- To build a machine learning model that will predict the likehood of drought occuring in the US for use in the Agricultural Sector.

Application of the Prediction Model

To improve drought resilience by:

- making different crop choices
- enrolling in crop insurance and other farm risk management
- investing in soil health and irrigation efficiency

Data Sources

Kaggle Data (https://www.kaggle.com/datasets/cdminix/us-drought-meteorological-data)

- Training Data(train_timeseries.csv) (2GB size, 2million records 2000-2016)
- Testing Data(test_timeseries.csv)(2019-2020)
- Geographical Data(soil.csv)

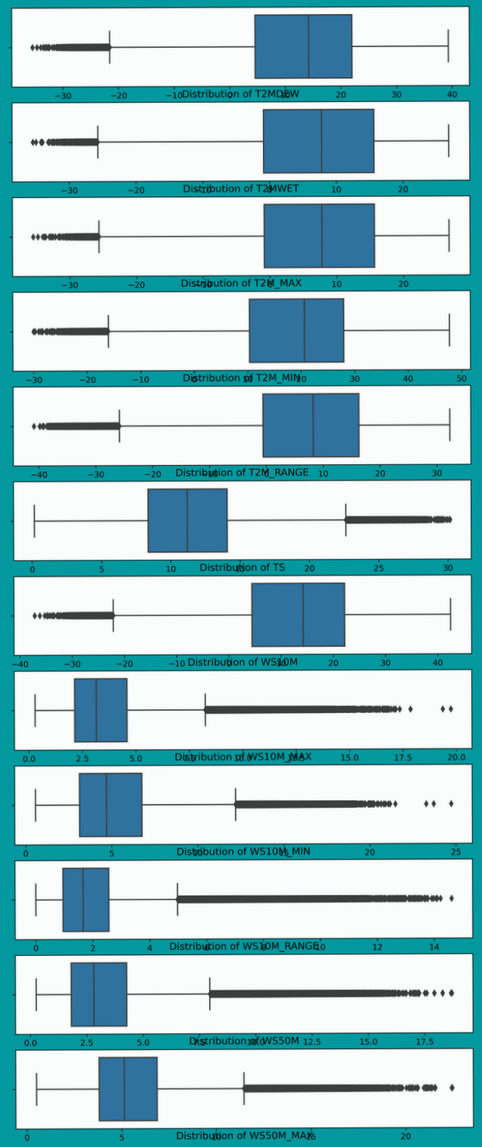
Original Source: NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program

Data Cleanup

- Import raw data csv files to jupyter notebook
- Remove the NaN values
- Change datatype

Data Exploration Analysis (EDA)

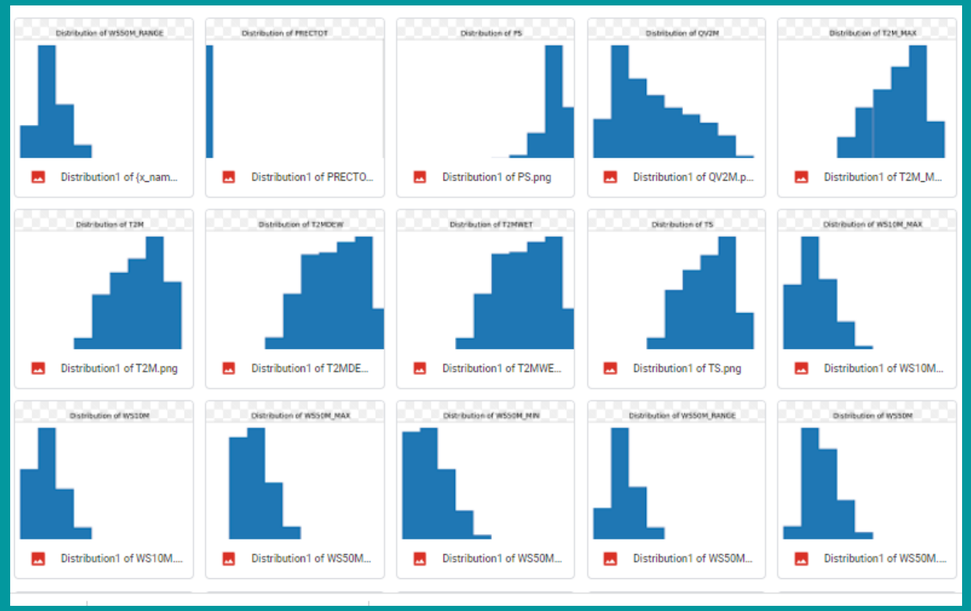
- Two methods for EDA
    - Skewness
    - Kurtosis
- Outlier Treatment and removing values beyond the standard outlier limit. First clean Data!



UNIVARIATE ANALYSIS

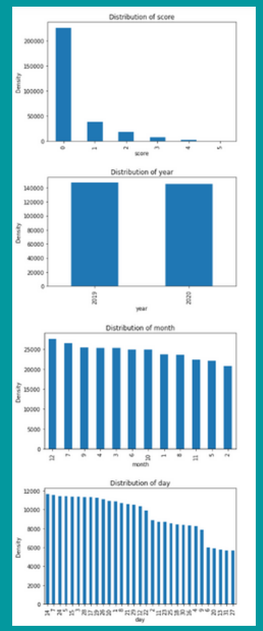
Continuous variables (all variables)

- Normal Distribution of 21 variables by plotting Histogram



Categorical variables(score, year, month and day)

- Normal to almost uniform distribution of 4 variables by plotting Histogram



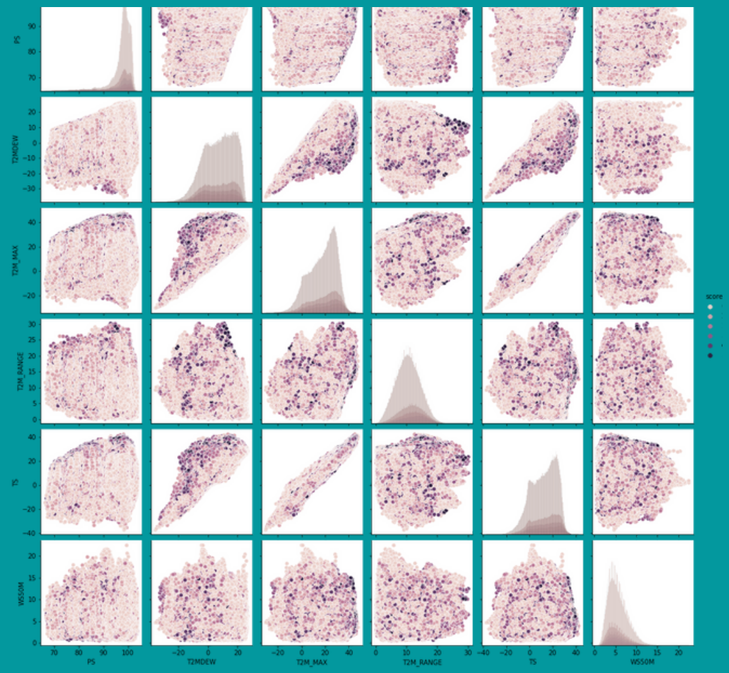
BIVARIATE ANALYSIS

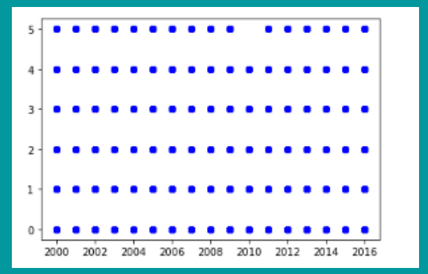
AGRICULTURE

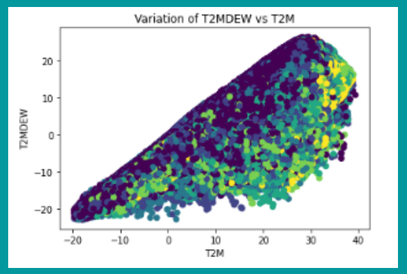
- Drought parameters for agriculture purpose and the impact of meteorological conditions.
- Looking for if main meteorological parameters such as precipitation, moisture, wind and temperature really contribute to ML
-

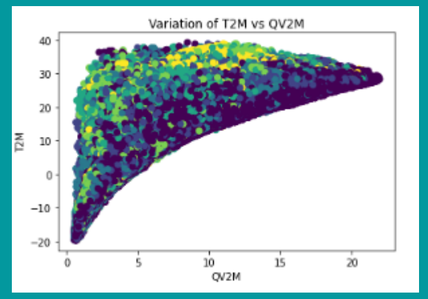
DATA ANALYSIS

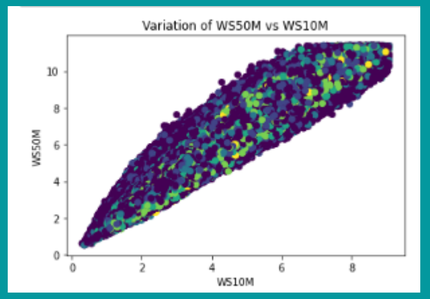
- Used Seaborn for scatter matrix to find correlation between dependent and independent variables
- Histogram and a lot of 2D scatter plots for the distribution along the diagnol





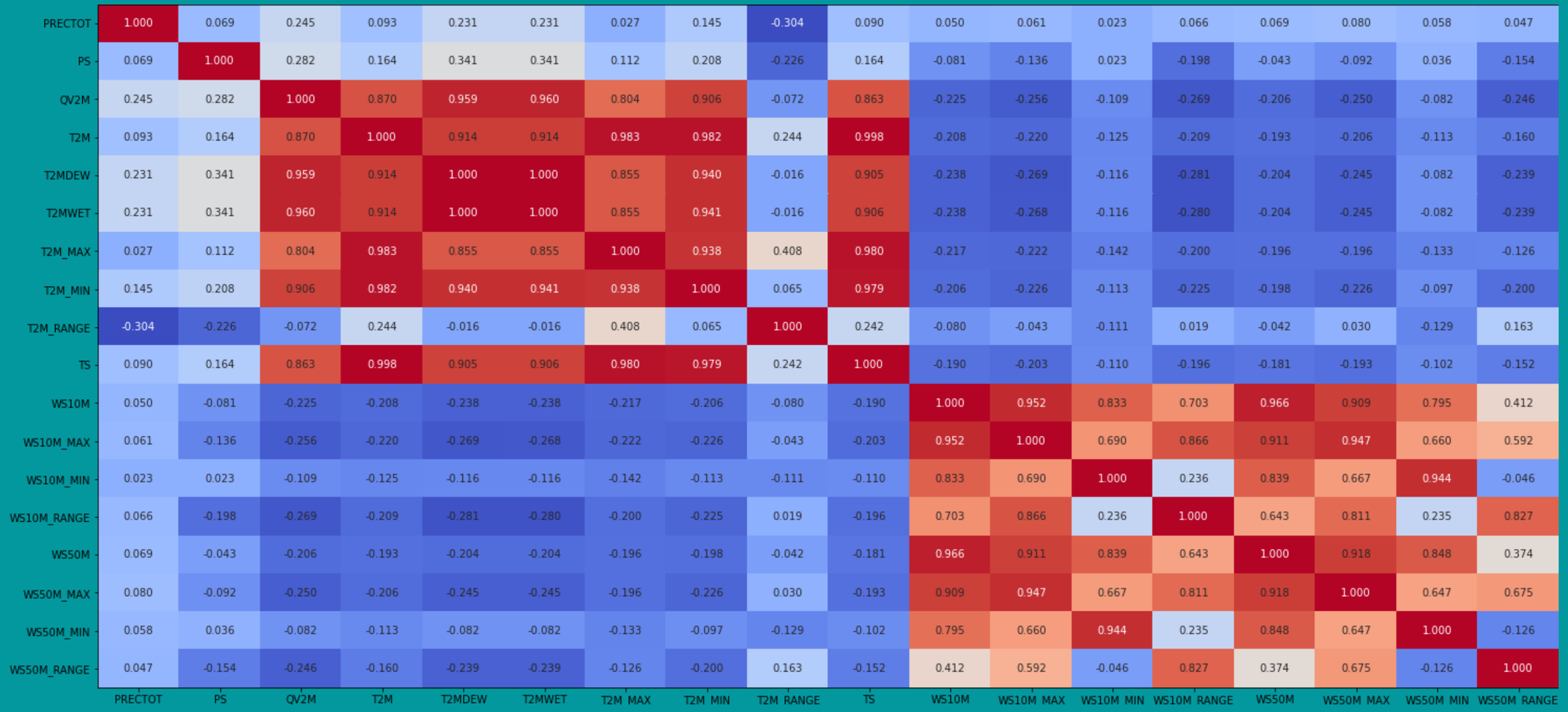






Correlation Heat Map

- To find correlation between independent variables for feature selection
- Values closer to 0 indicate no correlation whereas value closer to 1 indicate positive correlation between the variables



Data Modelling

Features Selection

- Used Random Forest Classifier model to selct the features
- The features are as follows:
  1. PS(Surface Pressure)
  2. T2MDEW(Dew Point at 2M)
  3. T2M_MAX(Maximum Temperature at 2M)
  4. T2M_RANGE(Temperature Range at 2M)
  5. TS(Earth Skin Temperature)
  6. WS50M(Wind Speed at 50M)

```
model = RandomForestClassifier(n_estimators=10) # n_estimators is the hyperparameter
rfe = RFE(model, n_features_to_select=6) # n_features_to_select is chosen on a trial and error basis
fit = rfe.fit(X_train, y_train)
print("Num Features: %s" % (fit.n_features_))
print("Selected Features: %s" % (fit.support_))
print("Feature Ranking: %s" % (fit.ranking_))
selected_features = independent_variables.columns[(fit.get_support())]
print(selected_features)
```

```
Num Features: 6
Selected Features: [False  True False False  True False  True False  True  True False False
 False False  True False False False False False False]
Feature Ranking: [13  1  5  8  1 11  1  4  1  1 14  2 16 10  1  9 12  7  6 15  3]
Index(['PS', 'T2MDEW', 'T2M_MAX', 'T2M_RANGE', 'TS', 'WS50M'], dtype='object')
```

Random Forest Algorithm without resampling - Setting the right Hyperparameters

In [59]:
```
RF_classifier = RandomForestClassifier(n_estimators = 50, max_depth=80, bootstrap=False, max_features='sqrt', random_state=0)
RF_classifier.fit(X_train, y_train)
y_pred_RF = RF_classifier.predict(X_test)
```

In [60]:
```
pickle.dump(RF_classifier, open('RF_classifier.pkl', 'wb'))
```

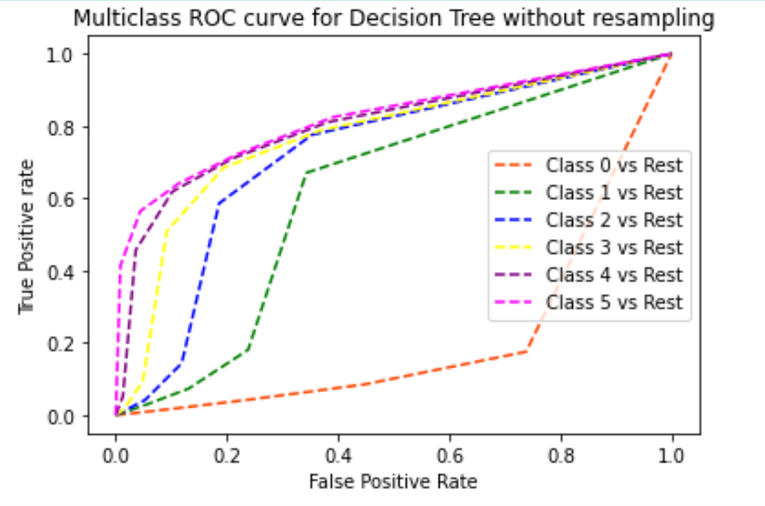
In [61]:
```
import gzip, pickle, pickletools
```

In [62]:
```
filepath = "RF_classifier.pkl"
with gzip.open(filepath, "wb") as f:
    pickled = pickle.dumps(RF_classifier)
    optimized_pickle = pickletools.optimize(pickled)
    f.write(optimized_pickle)
```

Data Modelling

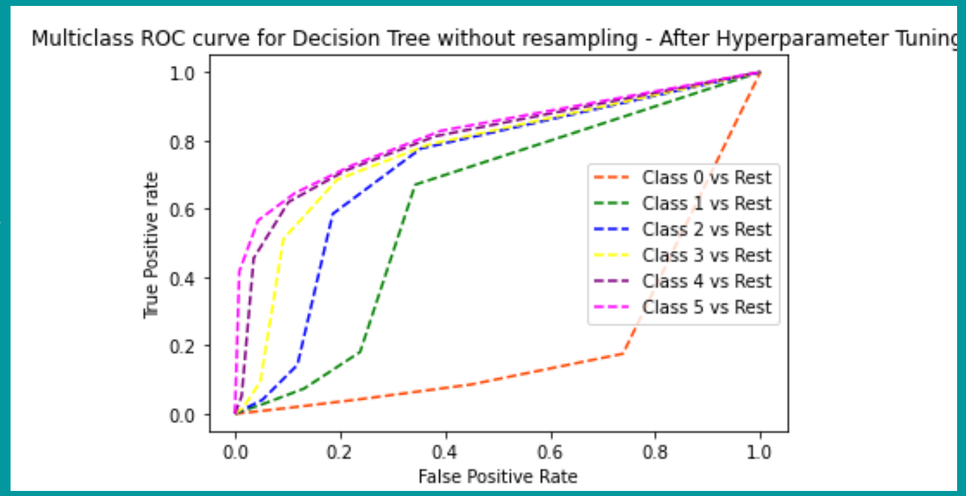
Model Development

Decision Tree Algorithm without resampling




```
Accuracy: 0.6777989281990349
Precision: 0.6788165543184689
Recall: 0.6777989281990349
F1 Score: 0.6783019240402571
Cohen Kappa Score: 0.4536519273494525
```

After Hyperparameter tuning

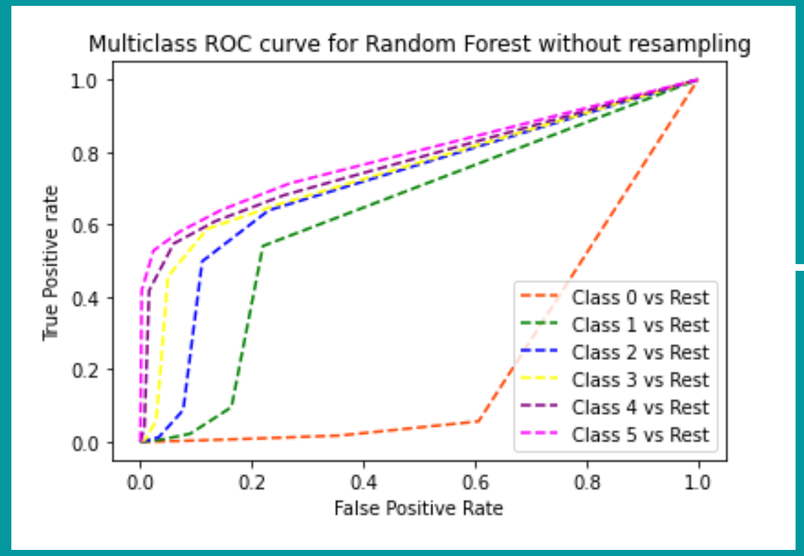



```
Accuracy: 0.6778898615388346
Precision: 0.6787938223854018
Recall: 0.6778898615388346
F1 Score: 0.6783368711313027
Cohen Kappa Score: 0.4536748664726149
```

Data Modelling

Model Development

Random Forest Algorithm without resampling



Accuracy: 0.7369015575870738
Precision: 0.7154562883084277
Recall: 0.7369015575870738
F1 Score: 0.7148504178166684
Cohen Kappa Score: 0.5019311370578048

After Hyperparameter tuning

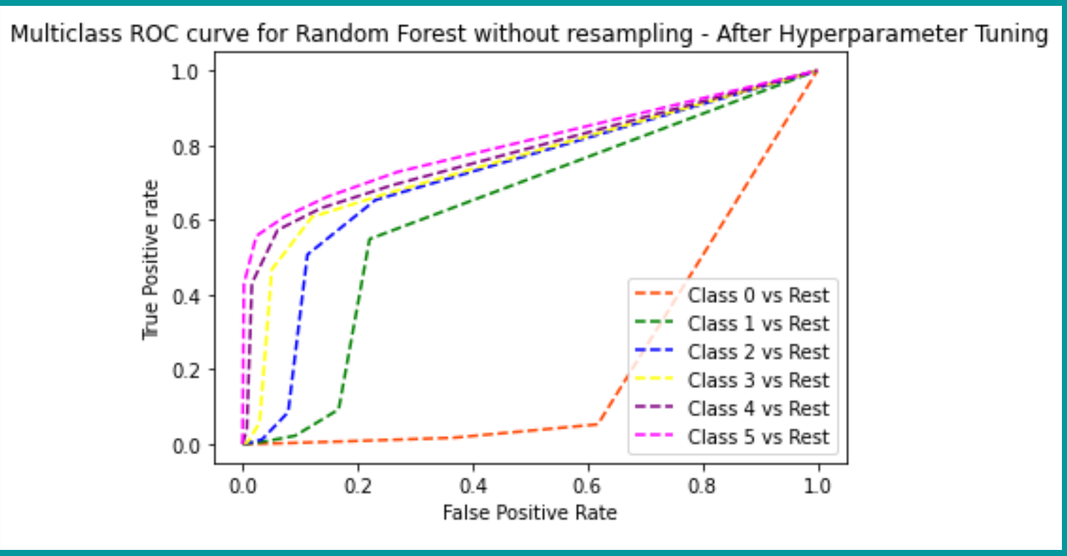
```
Performance of RF Algorithm without resampling - After Hyperparamter Tuning:

[[283544  10757   2663   1632    549    100]
 [ 37483  37892   5855   1327    410     99]
 [ 18215   7688  22177   3838    518    113]
 [ 11427   2109   4845  14137   1827    148]
 [  5658   1134   1076   2635   7228    689]
 [  1927    472    384    361    908   3043]]
              precision    recall  f1-score   support

           0       0.79      0.95      0.86    299245
           1       0.63      0.46      0.53     83066
           2       0.60      0.42      0.50     52549
           3       0.59      0.41      0.48     34493
           4       0.63      0.39      0.48     18420
           5       0.73      0.43      0.54      7095

    accuracy                           0.74    494868
   macro avg       0.66      0.51      0.57    494868
weighted avg       0.72      0.74      0.72    494868

Accuracy: 0.7436750810317094
Precision: 0.7232566901554951
Recall: 0.7436750810317094
F1 Score: 0.7225073834621258
Cohen Kappa Score: 0.5159807822270135
```



Accuracy: 0.7436750810317094
Precision: 0.7232566901554951
Recall: 0.7436750810317094
F1 Score: 0.7225073834621258
Cohen Kappa Score: 0.5159807822270135

Random Forest Algorithm was our choosen model for prediction based on accuracy.

Tools

1. PYTHON PANDAS
2. PYTHON MATPLOTLIB
3. SEABORN
4. NUMPY
5. HTML/CSS/BOOTSTRAP
6. PYTHON FLASK POWERED API
7. SCIKIT-LEARN AND PICKLE
8. GZIP
9. GITHUB AND HEROKU
10. AWS

Limitations

Size of train_timeseries.csv(16 years of daily data(more than 2mil rows)

- Capacity of computer was an issue(more than 32 GB run)
- Time Constraint(more than 3 hrs to develop and run model)

Model('pkl' file of size 6 GB)

- As Github's capacity was 4 GB, compressing of pkl file(600 MB) was done (3 hrs to run)
- Deployment to Heroku was a problem(supported size of 500 MB)
- Alternative cloud platform was AWS(limited time)

Thank you.